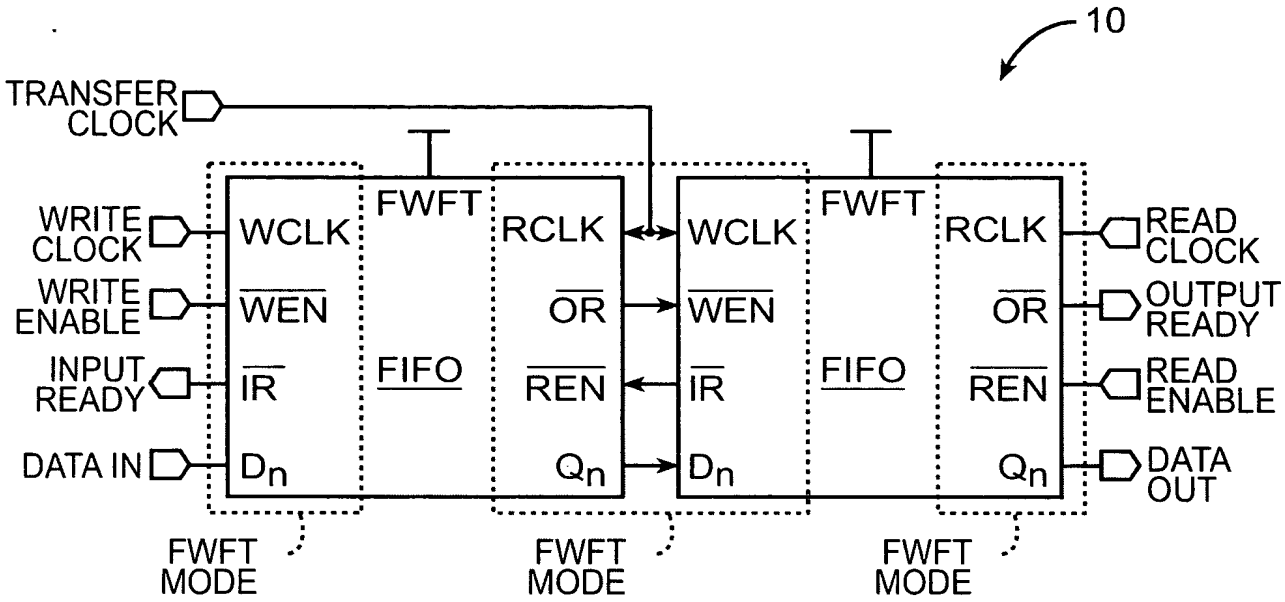



FIG. 1A
(PRIOR ART)

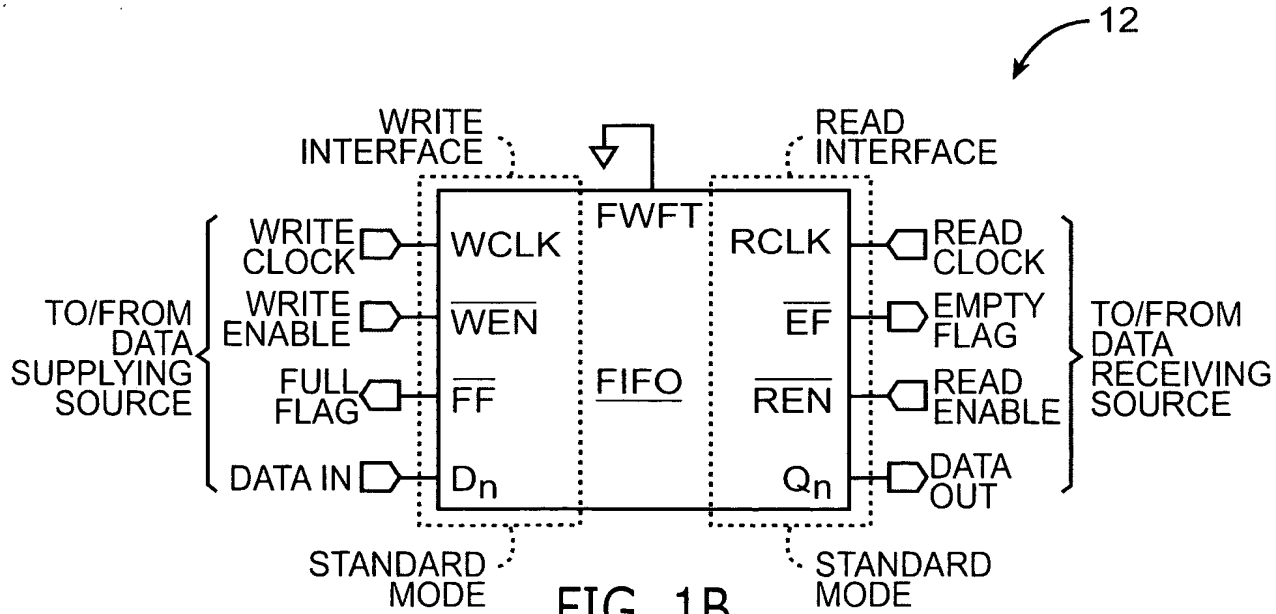



FIG. 1B
(PRIOR ART)

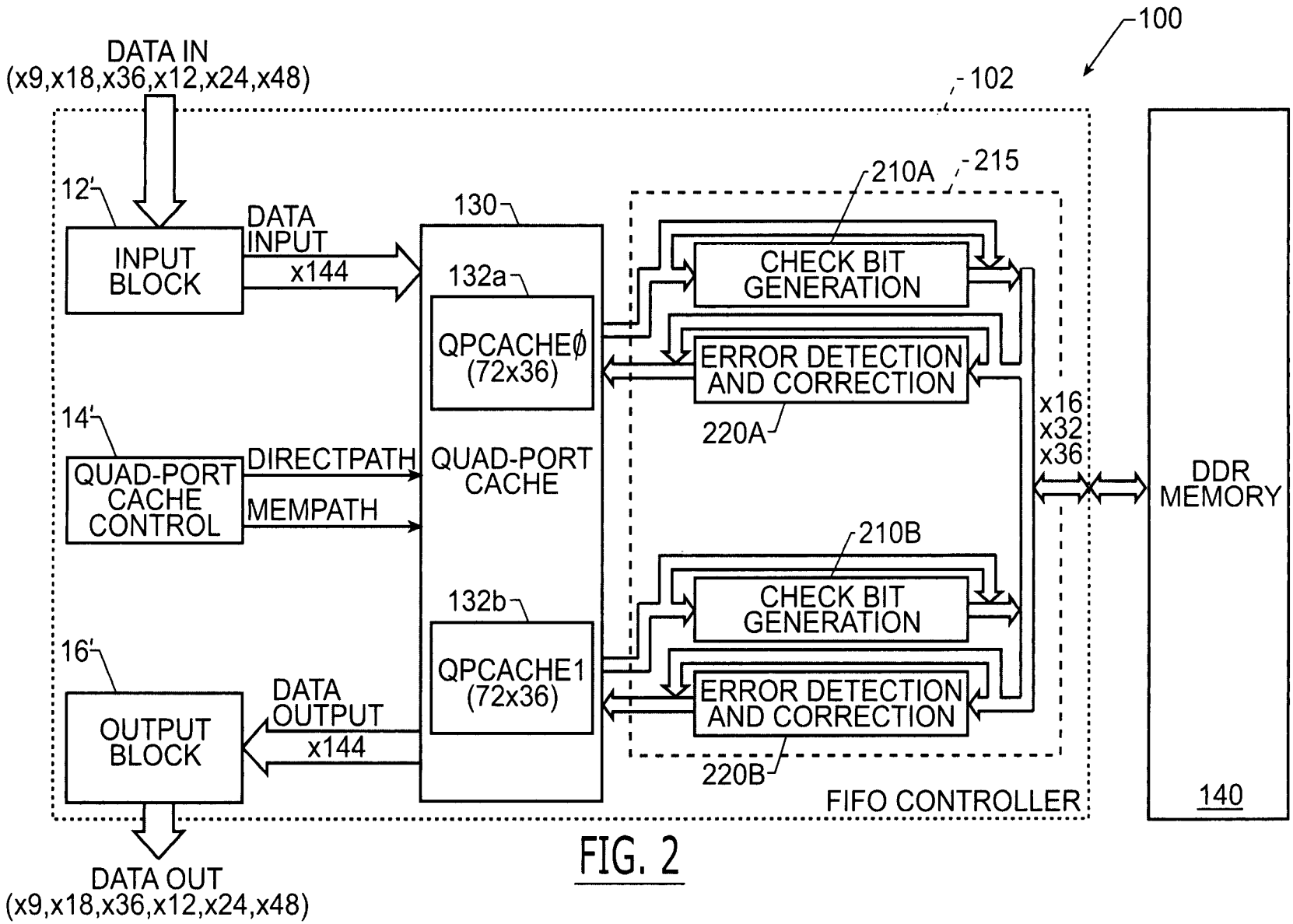

DATA IN
(x9,x18,x36,x12,x24,x48)
100
102
12'
INPUT BLOCK
DATA INPUT
x144
130
132a
QPCACHEØ
(72x36)
215
210A
CHECK BIT GENERATION
ERROR DETECTION AND CORRECTION
220A
14'
QUAD-PORT CACHE CONTROL
DIRECTPATH
MEMPATH
QUAD-PORT CACHE
x16
x32
x36
DDR MEMORY
210B
132b
CHECK BIT GENERATION
16'
OUTPUT BLOCK
DATA OUTPUT
x144
QPCACHE1
(72x36)
ERROR DETECTION AND CORRECTION
220B
FIFO CONTROLLER
140
DATA OUT
(x9,x18,x36,x12,x24,x48)

FIG. 2

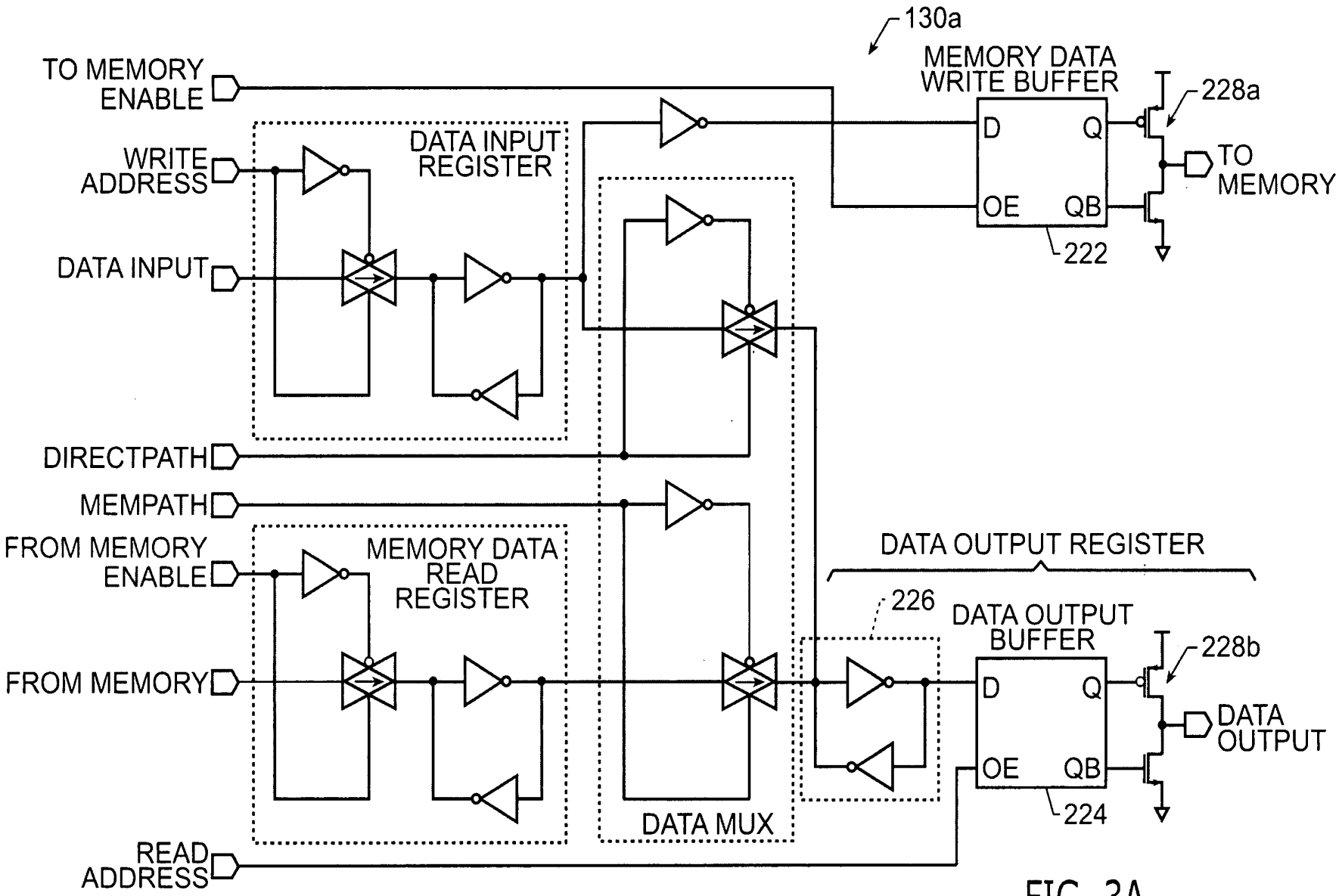

130a
TO MEMORY ENABLE
WRITE ADDRESS
DATA INPUT
DIRECTPATH
MEMPATH
FROM MEMORY ENABLE
FROM MEMORY
READ ADDRESS
DATA INPUT REGISTER
MEMORY DATA READ REGISTER
DATA MUX
MEMORY DATA WRITE BUFFER
D
Q
OE
QB
222
228a
TO MEMORY
DATA OUTPUT REGISTER
226
DATA OUTPUT BUFFER
224
228b
DATA OUTPUT

FIG. 3A

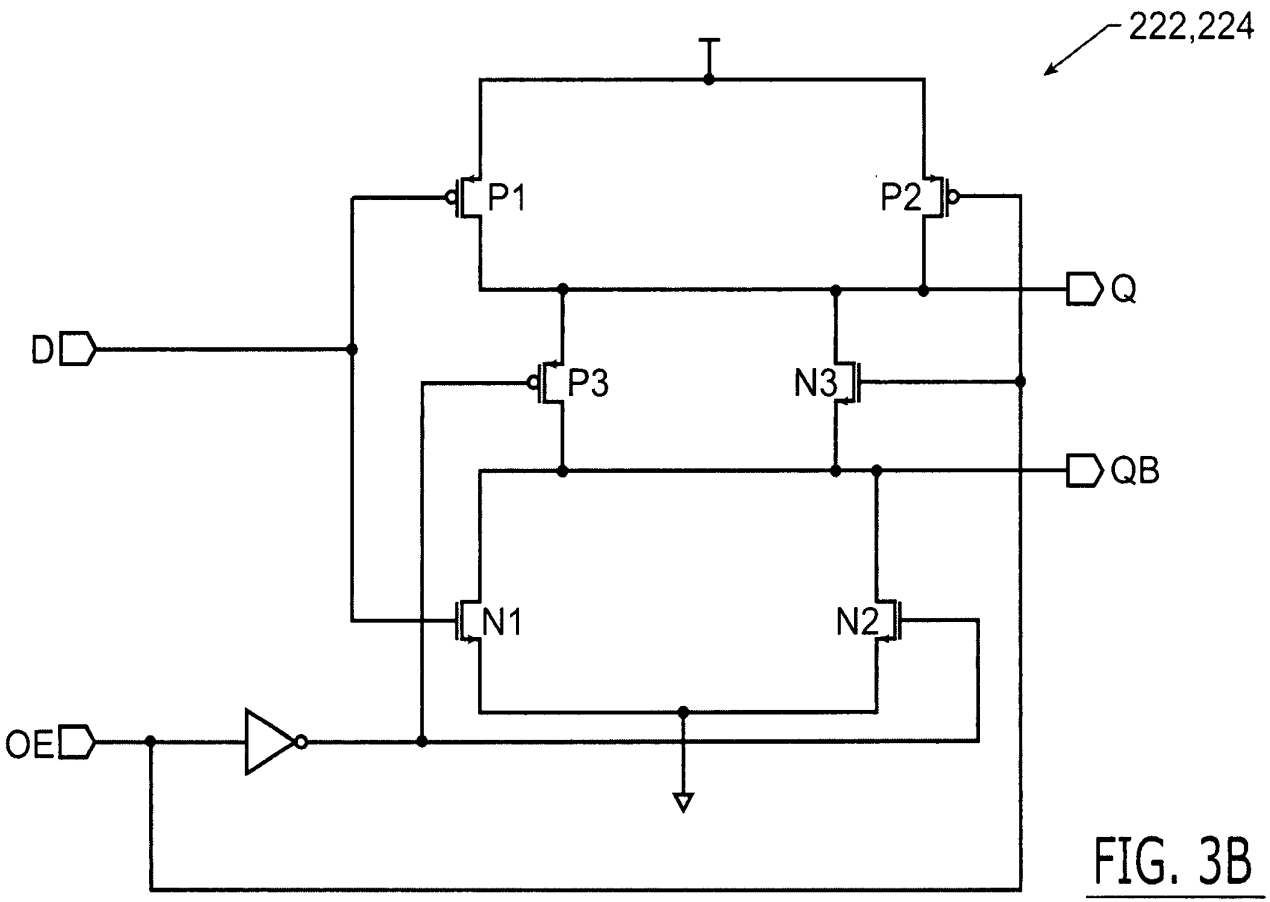



FIG. 3B

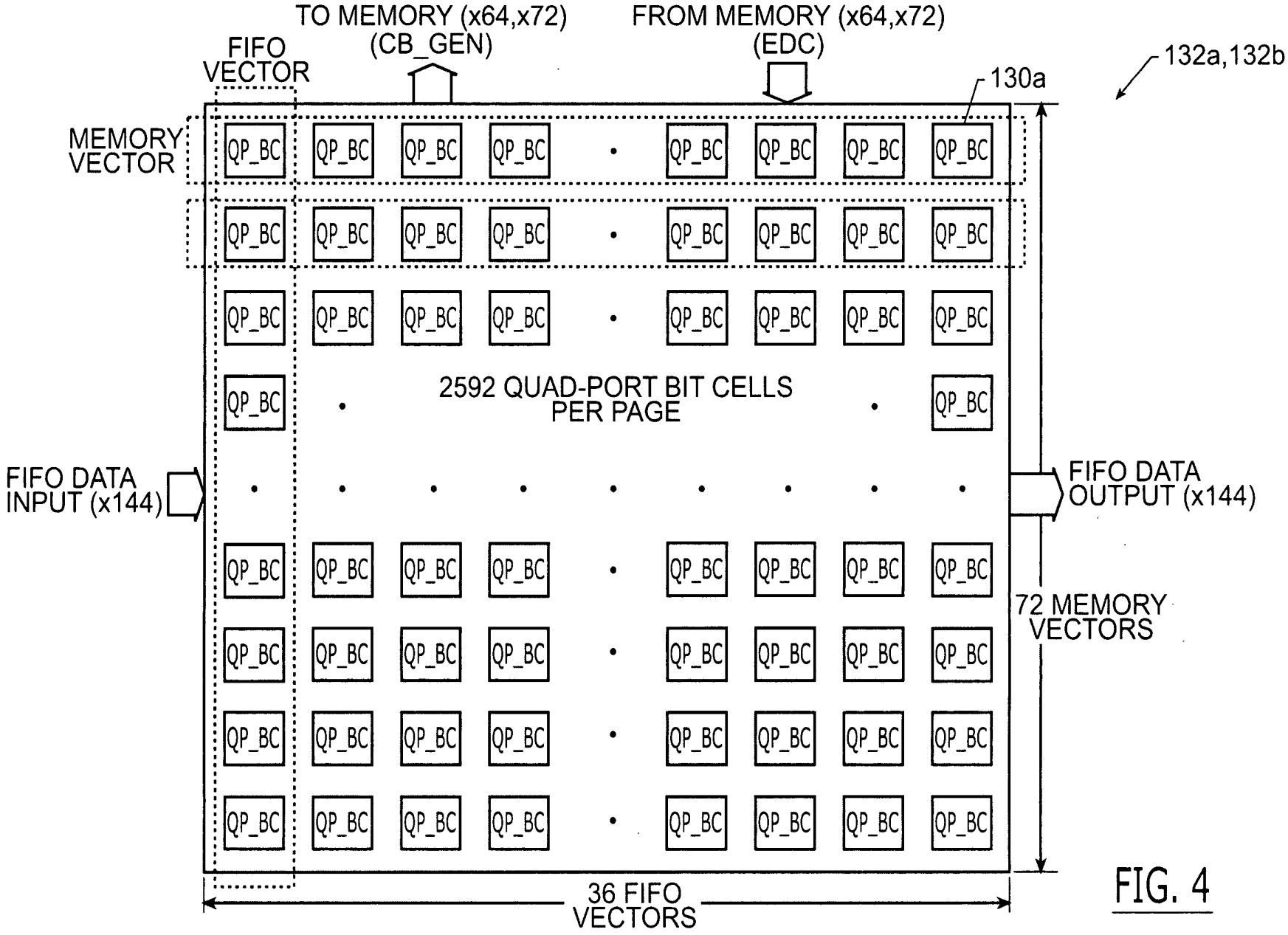

TO MEMORY (x64,x72)
(CB_GEN)
FROM MEMORY (x64,x72)
(EDC)
FIFO
VECTOR
132a,132b
130a
MEMORY
VECTOR
QP_BC
2592 QUAD-PORT BIT CELLS
PER PAGE
FIFO DATA
INPUT (x144)
FIFO DATA
OUTPUT (x144)
72 MEMORY
VECTORS
36 FIFO
VECTORS

FIG. 4

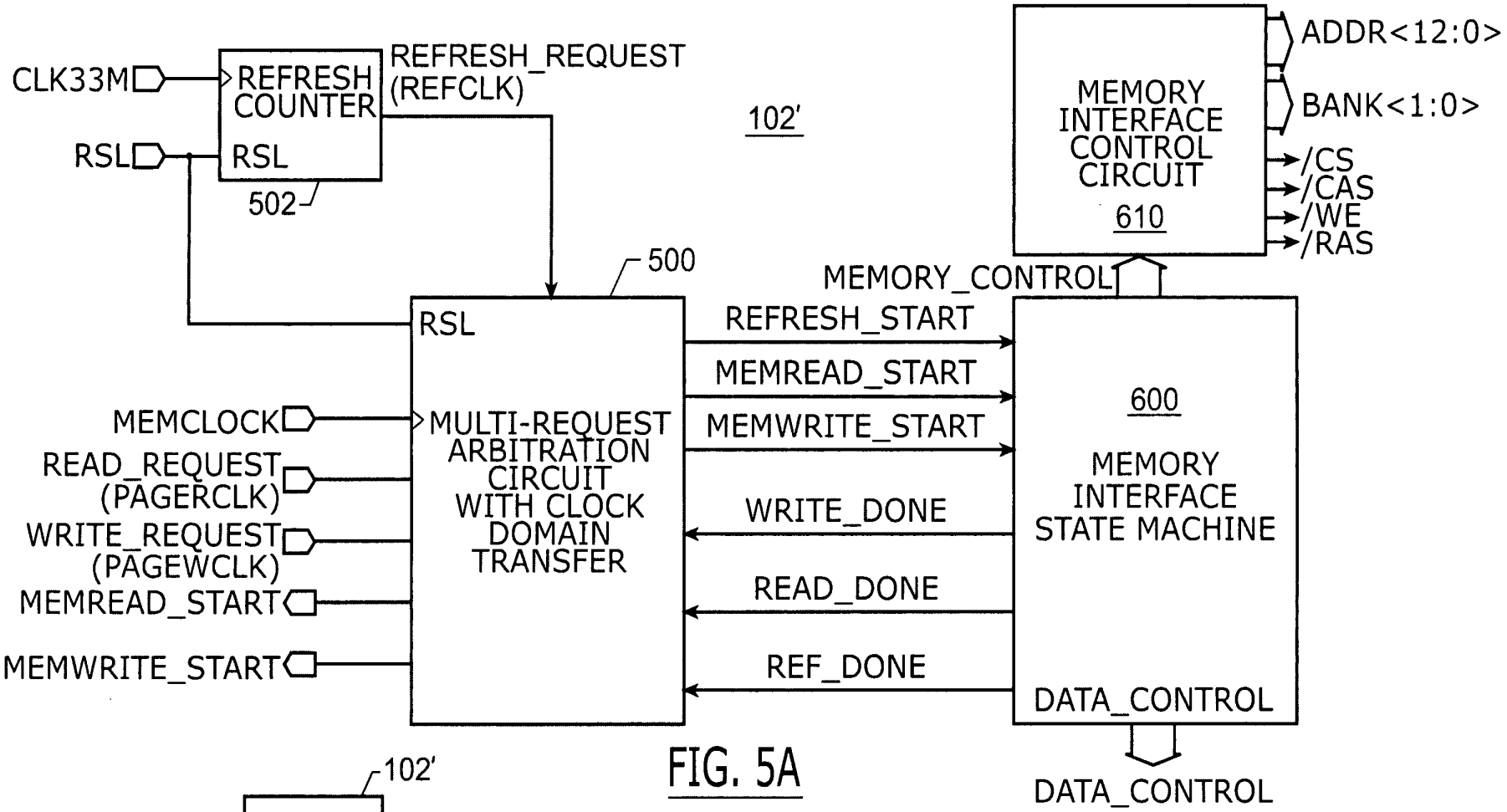

CLK33M
REFRESH COUNTER
REFRESH_REQUEST (REFCLK)
RSL
RSL
502
102'
MEMORY INTERFACE CONTROL CIRCUIT
610
ADDR<12:0>
BANK<1:0>
/CS
/CAS
/WE
/RAS
500
MEMORY_CONTROL
RSL
REFRESH_START
MEMREAD_START
MEMWRITE_START
MEMCLOCK
READ_REQUEST (PAGERCLK)
WRITE_REQUEST (PAGEWCLK)
MEMREAD_START
MEMWRITE_START
MULTI-REQUEST ARBITRATION CIRCUIT WITH CLOCK DOMAIN TRANSFER
WRITE_DONE
READ_DONE
REF_DONE
600
MEMORY INTERFACE STATE MACHINE
DATA_CONTROL
DATA_CONTROL

FIG. 5A

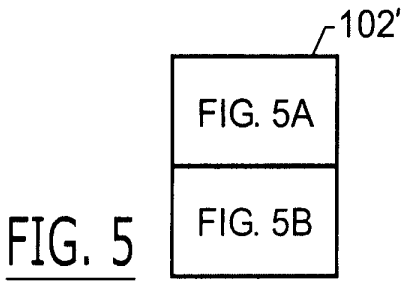

102'
FIG. 5A
FIG. 5B

FIG. 5

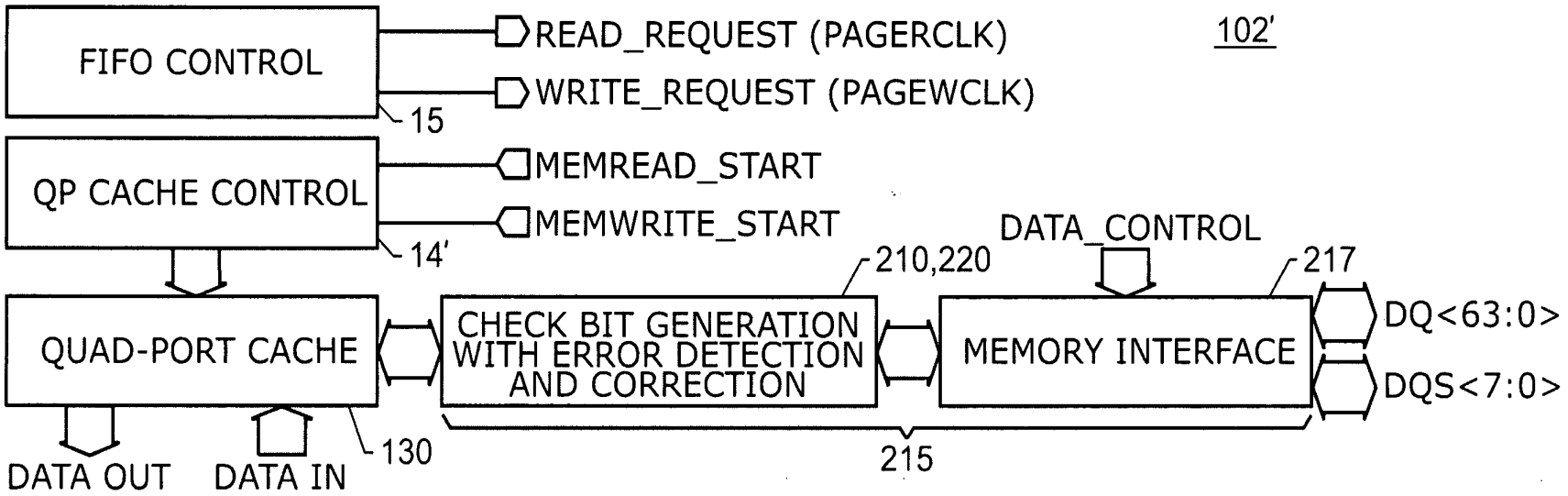
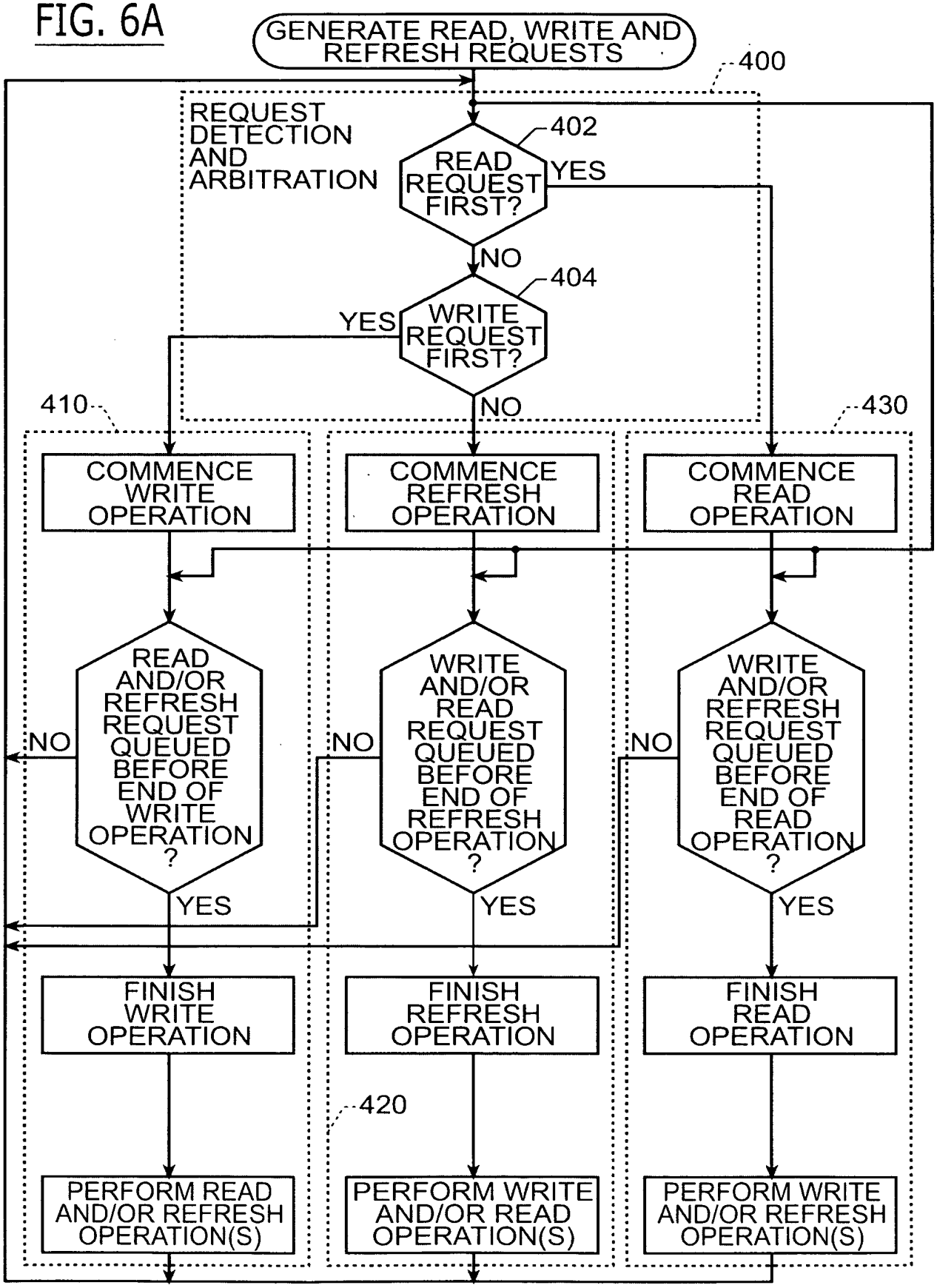

FIFO CONTROL
READ_REQUEST (PAGERCLK)
WRITE_REQUEST (PAGEWCLK)
102'
15
QP CACHE CONTROL
MEMREAD_START
MEMWRITE_START
14'
DATA_CONTROL
210,220
217
QUAD-PORT CACHE
CHECK BIT GENERATION WITH ERROR DETECTION AND CORRECTION
MEMORY INTERFACE
DQ<63:0>
DQS<7:0>
DATA OUT
DATA IN
130
215

FIG. 5B

FIG. 6A


GENERATE READ, WRITE AND REFRESH REQUESTS
400
REQUEST DETECTION AND ARBITRATION
402
READ REQUEST FIRST?
YES
NO
404
WRITE REQUEST FIRST?
YES
NO
410
430
COMMENCE WRITE OPERATION
COMMENCE REFRESH OPERATION
COMMENCE READ OPERATION
READ AND/OR REFRESH REQUEST QUEUED BEFORE END OF WRITE OPERATION ?
WRITE AND/OR READ REQUEST QUEUED BEFORE END OF REFRESH OPERATION ?
WRITE AND/OR REFRESH REQUEST QUEUED BEFORE END OF READ OPERATION ?
NO
NO
NO
YES
YES
YES
FINISH WRITE OPERATION
FINISH REFRESH OPERATION
FINISH READ OPERATION
420
PERFORM READ AND/OR REFRESH OPERATION(S)
PERFORM WRITE AND/OR READ OPERATION(S)
PERFORM WRITE AND/OR REFRESH OPERATION(S)

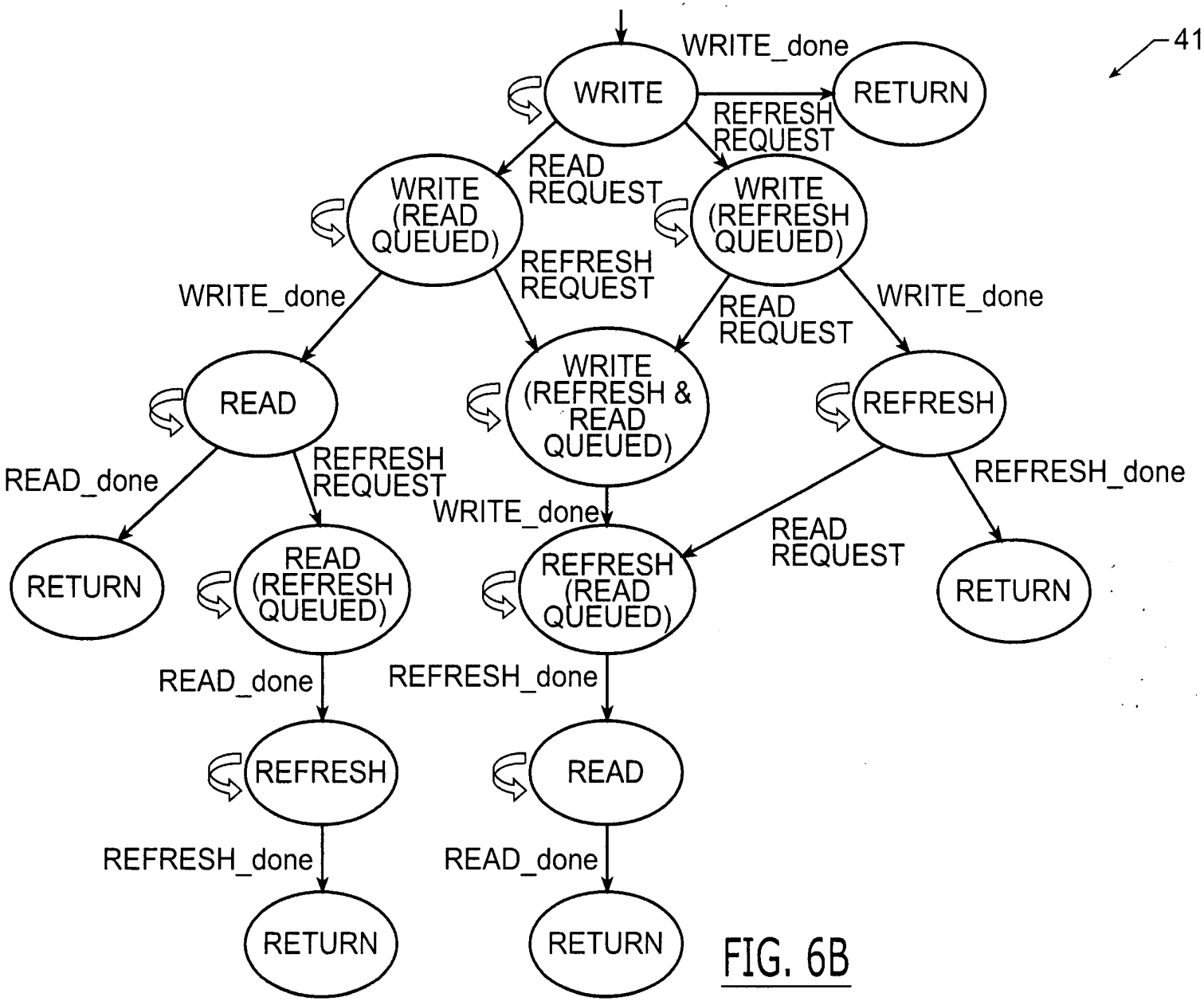

410
WRITE_done
WRITE
RETURN
REFRESH REQUEST
READ REQUEST
WRITE (READ QUEUED)
WRITE (REFRESH QUEUED)
REFRESH REQUEST
WRITE_done
READ REQUEST
WRITE_done
READ
WRITE (REFRESH & READ QUEUED)
REFRESH
READ_done
REFRESH REQUEST
REFRESH_done
WRITE_done
READ REQUEST
RETURN
READ (REFRESH QUEUED)
REFRESH (READ QUEUED)
RETURN
READ_done
REFRESH_done
REFRESH
READ
REFRESH_done
READ_done
RETURN
RETURN

FIG. 6B

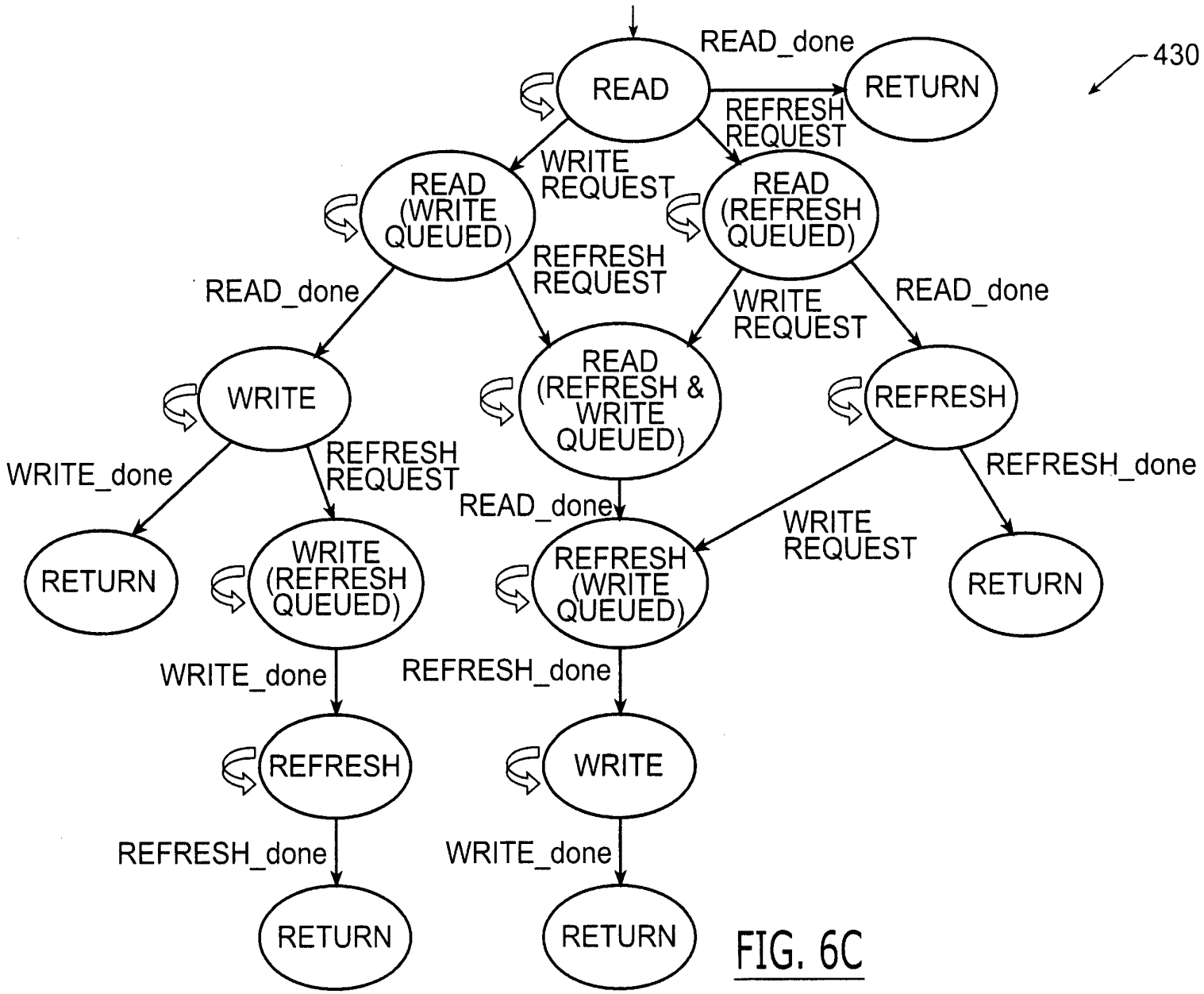

430
READ
READ_done
RETURN
REFRESH REQUEST
WRITE REQUEST
READ (WRITE QUEUED)
READ (REFRESH QUEUED)
REFRESH REQUEST
READ_done
WRITE REQUEST
READ_done
WRITE
READ (REFRESH & WRITE QUEUED)
REFRESH
WRITE_done
REFRESH REQUEST
READ_done
WRITE REQUEST
REFRESH_done
RETURN
WRITE (REFRESH QUEUED)
REFRESH (WRITE QUEUED)
RETURN
WRITE_done
REFRESH_done
REFRESH
WRITE
REFRESH_done
WRITE_done
RETURN
RETURN

FIG. 6C

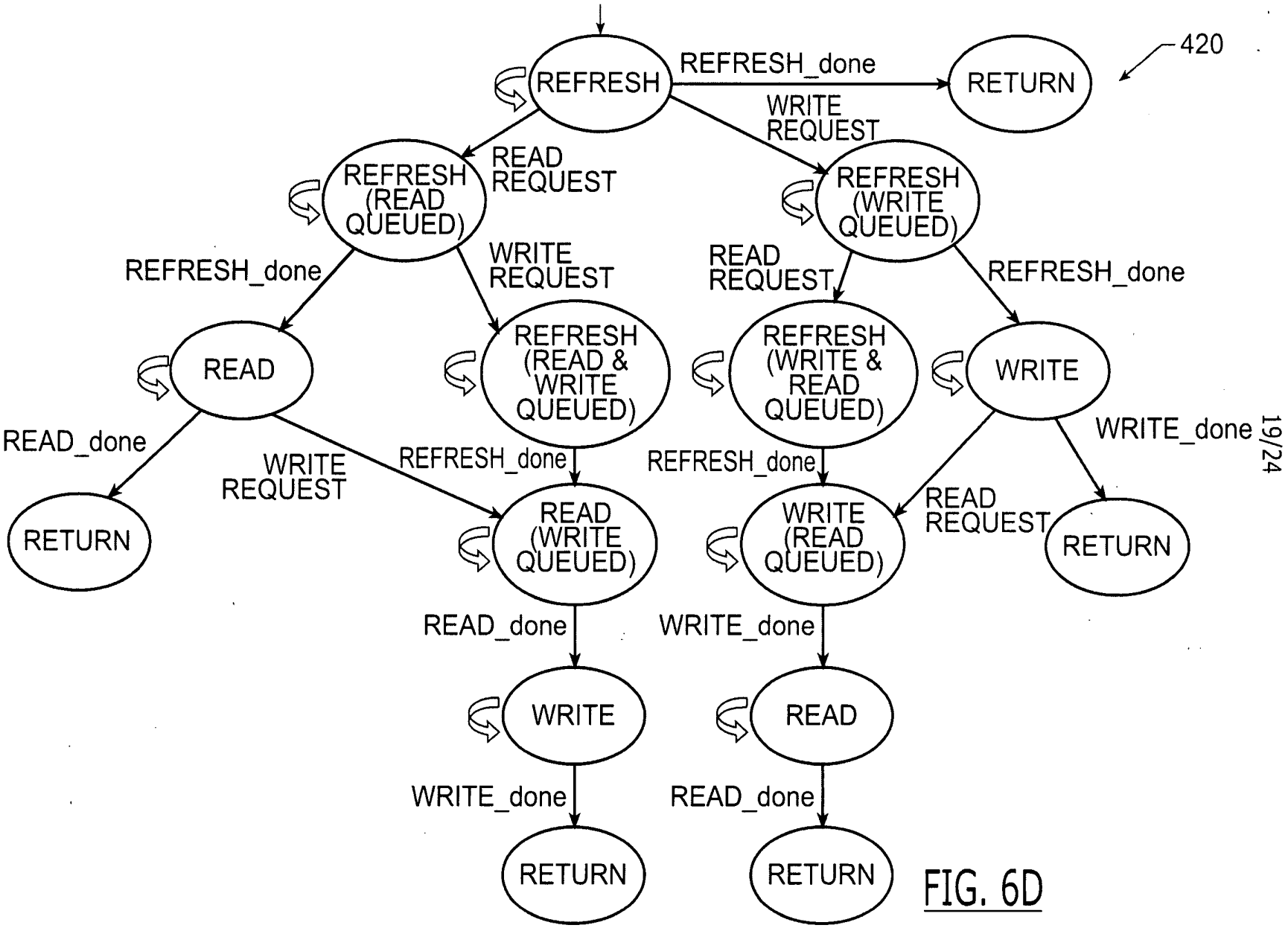

420
REFRESH
REFRESH_done
RETURN
WRITE REQUEST
READ REQUEST
REFRESH (READ QUEUED)
REFRESH (WRITE QUEUED)
REFRESH_done
WRITE REQUEST
READ REQUEST
REFRESH_done
READ
REFRESH (READ & WRITE QUEUED)
REFRESH (WRITE & READ QUEUED)
WRITE
READ_done
WRITE REQUEST
REFRESH_done
REFRESH_done
WRITE_done
RETURN
READ (WRITE QUEUED)
WRITE (READ QUEUED)
READ REQUEST
RETURN
READ_done
WRITE_done
WRITE
READ
WRITE_done
READ_done
RETURN
RETURN

FIG. 6D

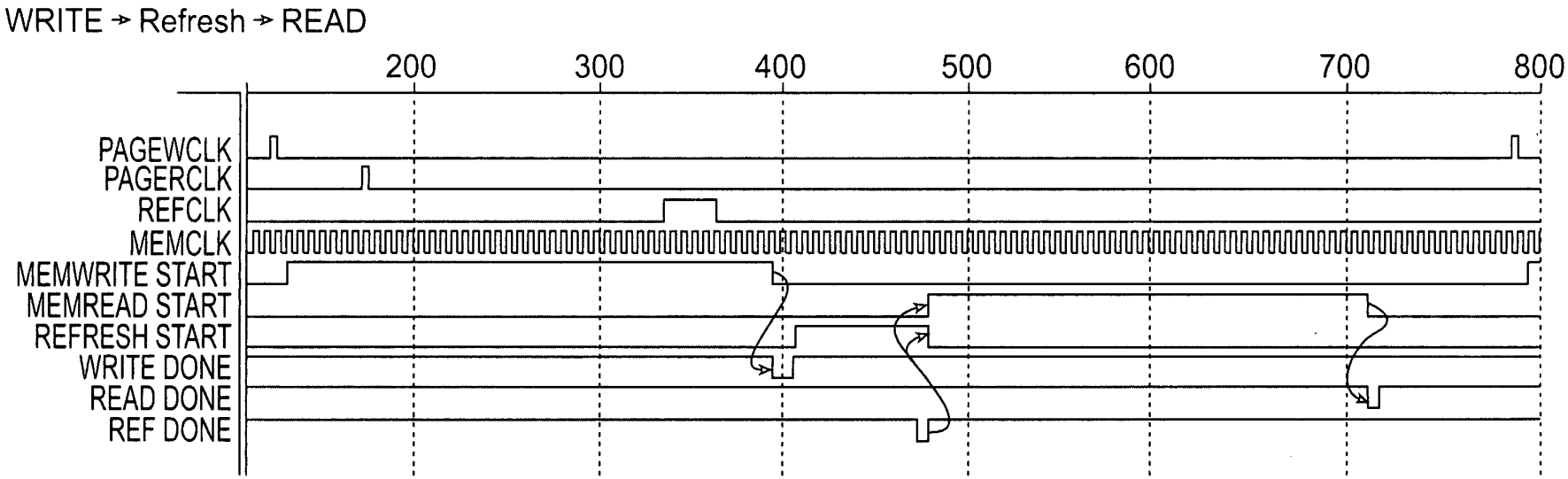

WRITE → Refresh → READ
200
300
400
500
600
700
800
PAGEWCLK
PAGERCLK
REFCLK
MEMCLK
MEMWRITE START
MEMREAD START
REFRESH START
WRITE DONE
READ DONE
REF DONE

FIG. 7A

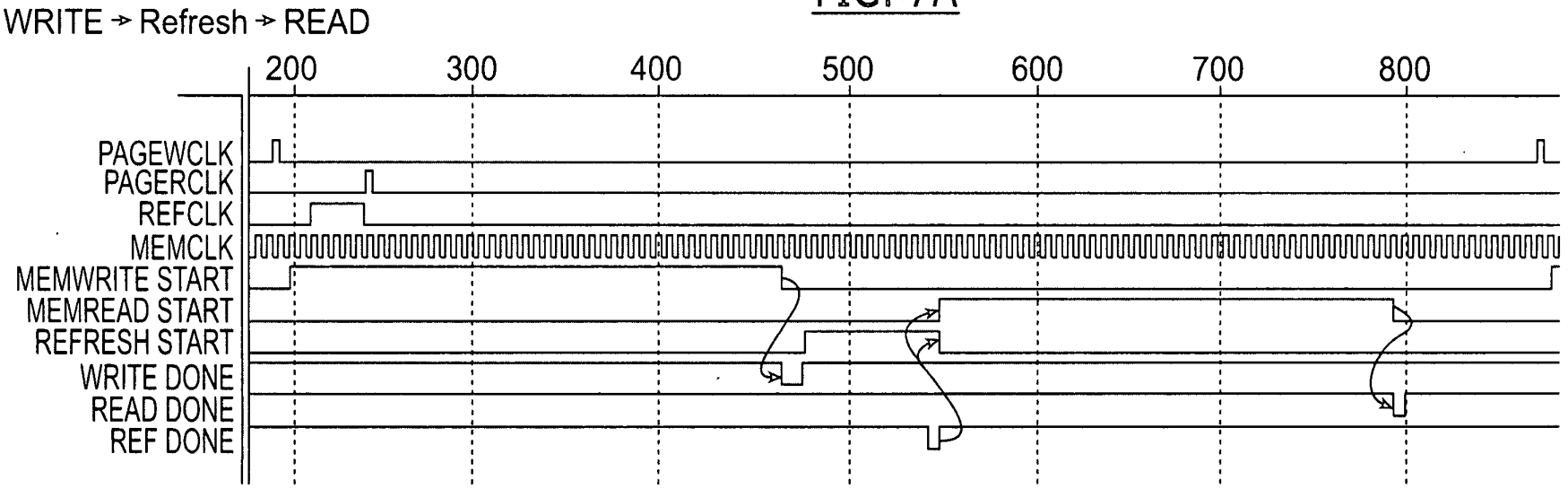

WRITE → Refresh → READ
200
300
400
500
600
700
800
PAGEWCLK
PAGERCLK
REFCLK
MEMCLK
MEMWRITE START
MEMREAD START
REFRESH START
WRITE DONE
READ DONE
REF DONE

FIG. 7B

WRITE → READ → Refresh

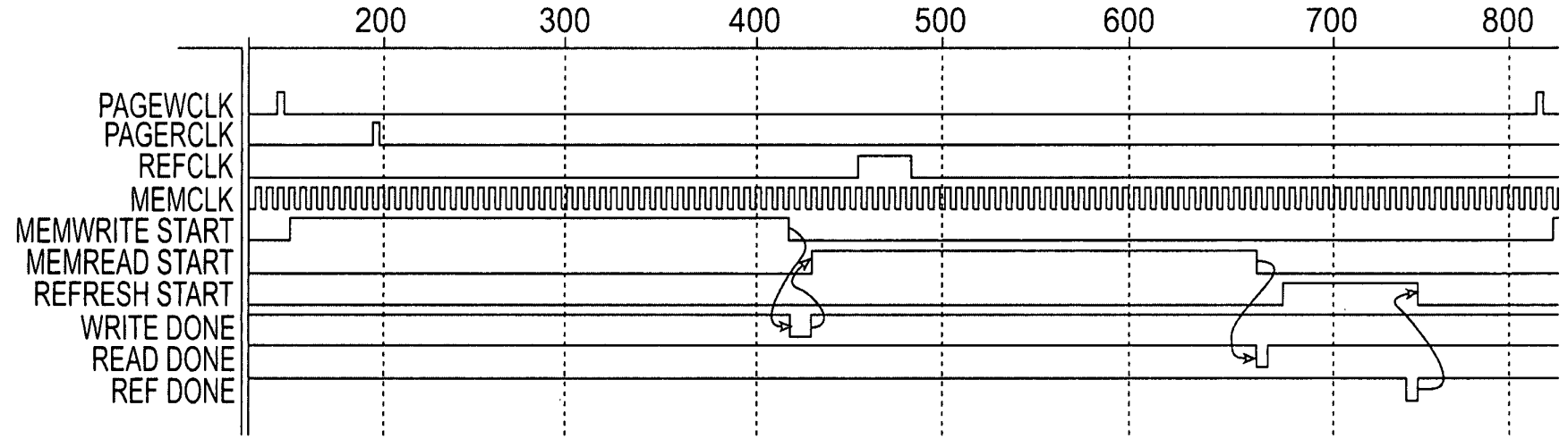



FIG. 7C

READ → Refresh → WRITE

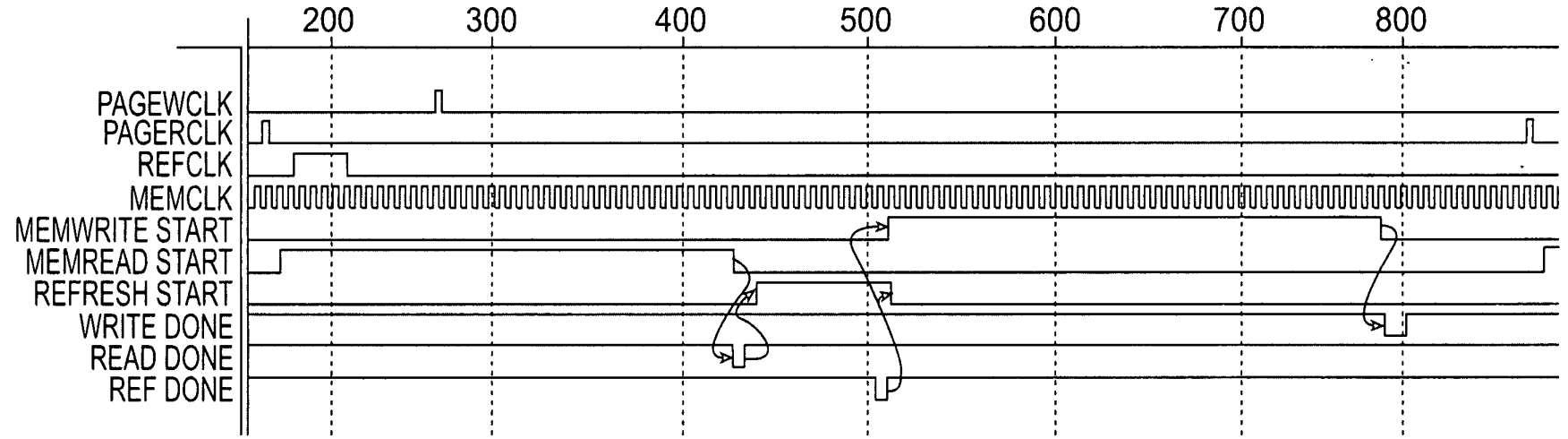



FIG. 7D

READ → Refresh → WRITE

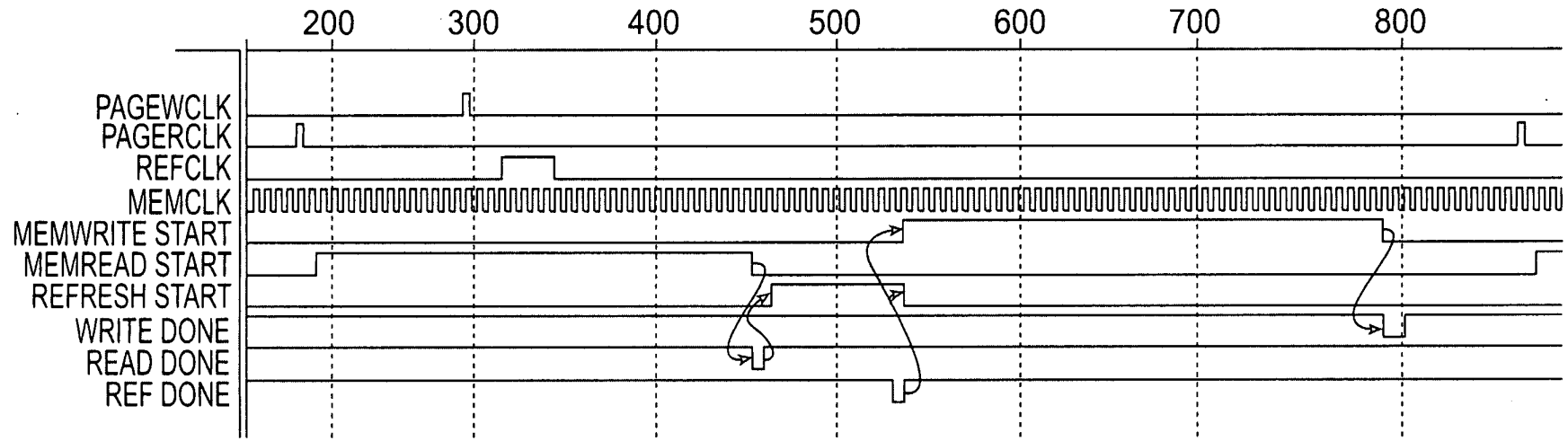



FIG. 7E

READ → WRITE → Refresh

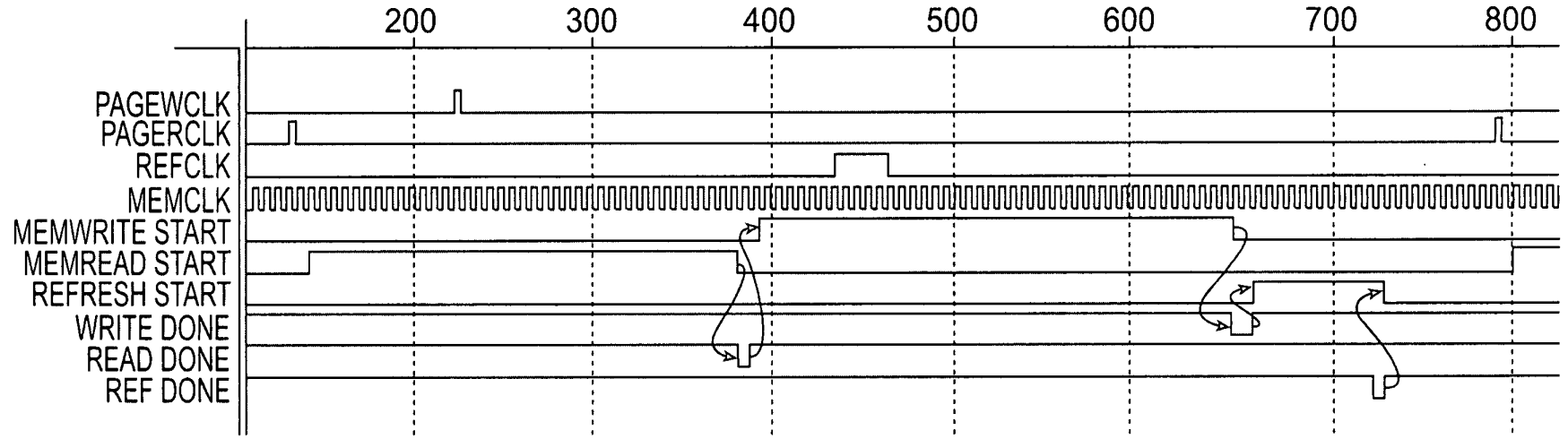



FIG. 7F

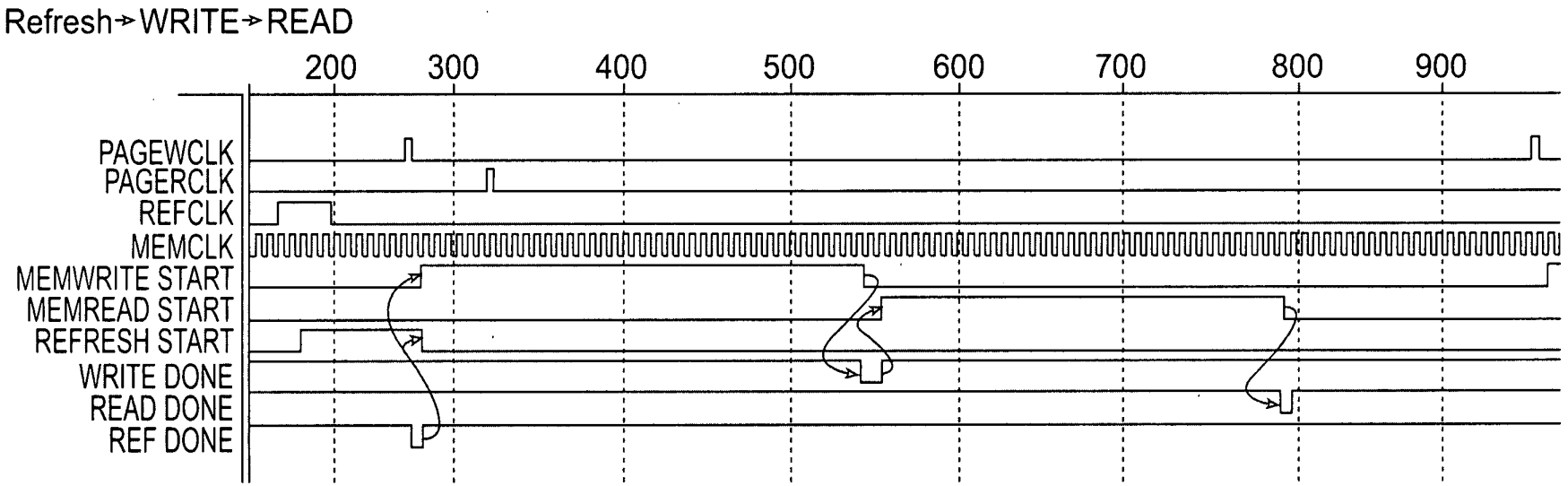

Refresh→WRITE→READ
200
300
400
500
600
700
800
900
PAGEWCLK
PAGERCLK
REFCLK
MEMCLK
MEMWRITE START
MEMREAD START
REFRESH START
WRITE DONE
READ DONE
REF DONE

FIG. 7G

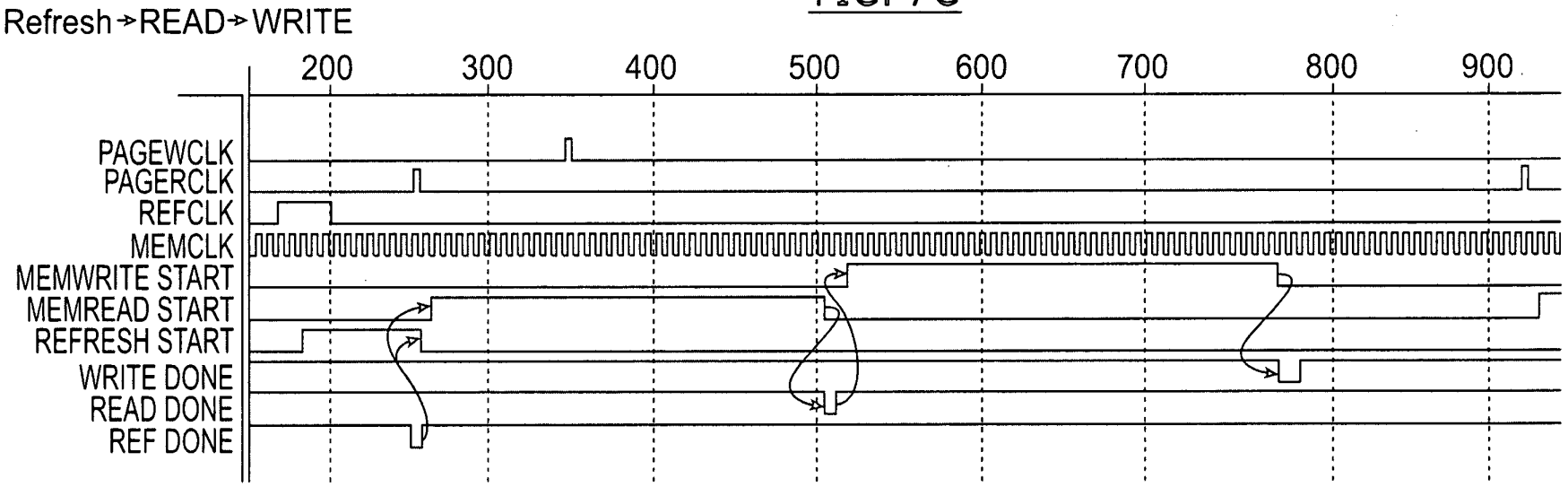

Refresh→READ→WRITE
200
300
400
500
600
700
800
900
PAGEWCLK
PAGERCLK
REFCLK
MEMCLK
MEMWRITE START
MEMREAD START
REFRESH START
WRITE DONE
READ DONE
REF DONE

FIG. 7H

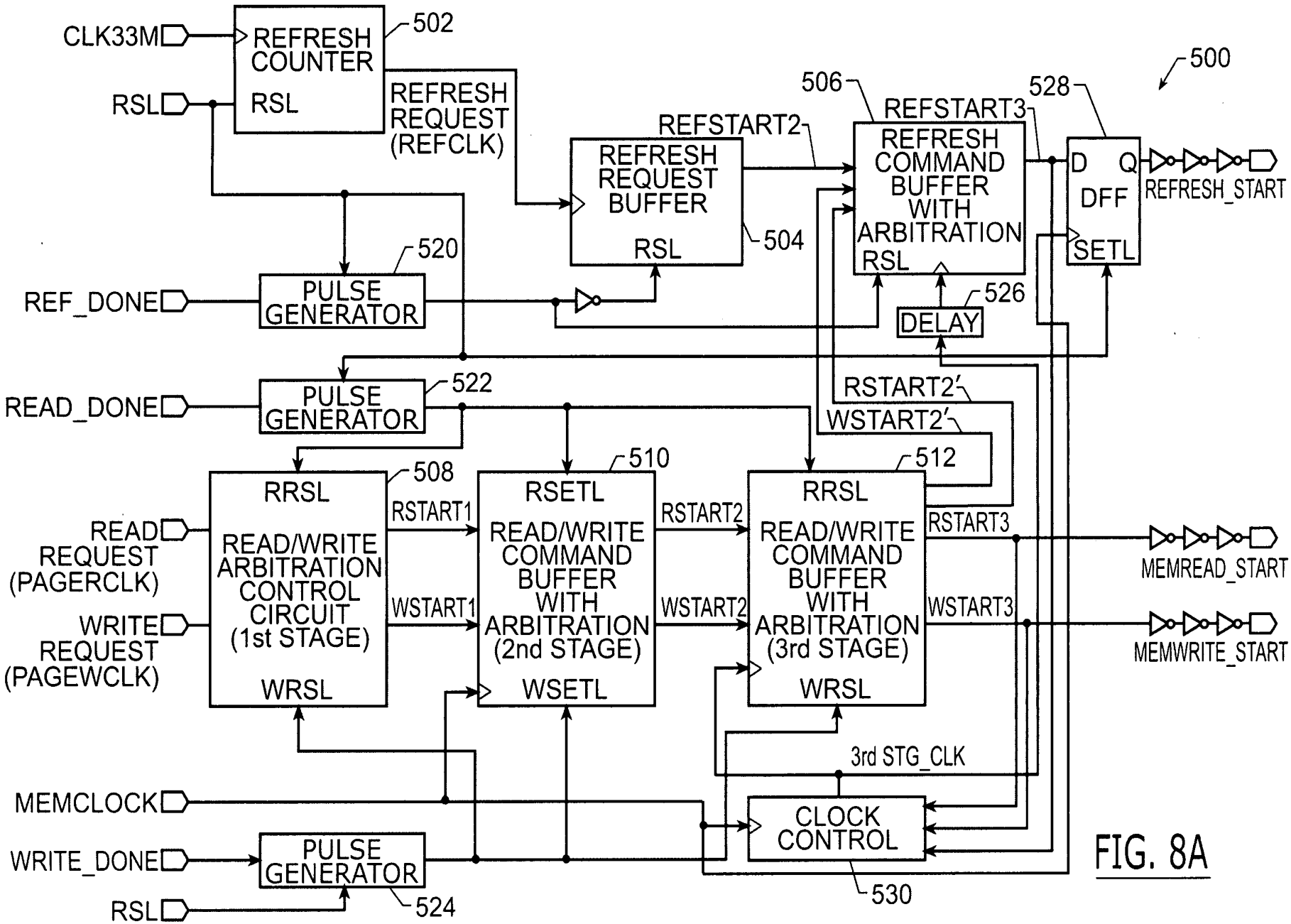

CLK33M
RSL
REFRESH COUNTER
RSL
502
REFRESH REQUEST (REFCLK)
REFRESH REQUEST BUFFER
RSL
504
REFSTART2
506
REFSTART3
REFRESH COMMAND BUFFER WITH ARBITRATION
RSL
528
D
Q
DFF
SETL
REFRESH_START
500
520
REF_DONE
PULSE GENERATOR
526
DELAY
522
READ_DONE
PULSE GENERATOR
RSTART2'
WSTART2'
508
510
512
RRSL
READ/WRITE ARBITRATION CONTROL CIRCUIT (1st STAGE)
WRSL
RSTART1
WSTART1
RSETL
READ/WRITE COMMAND BUFFER WITH ARBITRATION (2nd STAGE)
WSETL
RSTART2
WSTART2
RRSL
READ/WRITE COMMAND BUFFER WITH ARBITRATION (3rd STAGE)
WRSL
RSTART3
WSTART3
READ REQUEST (PAGERCLK)
WRITE REQUEST (PAGEWCLK)
MEMREAD_START
MEMWRITE_START
3rd STG_CLK
MEMCLOCK
CLOCK CONTROL
WRITE_DONE
PULSE GENERATOR
RSL
524
530

FIG. 8A

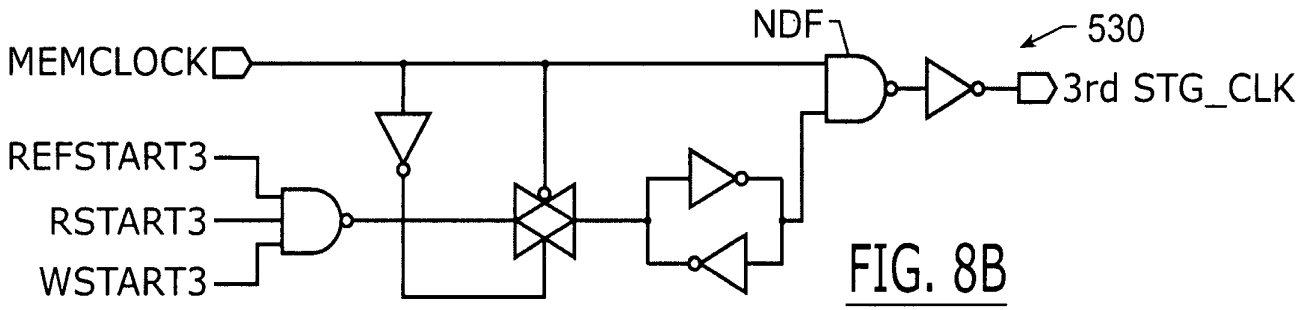



FIG. 8B

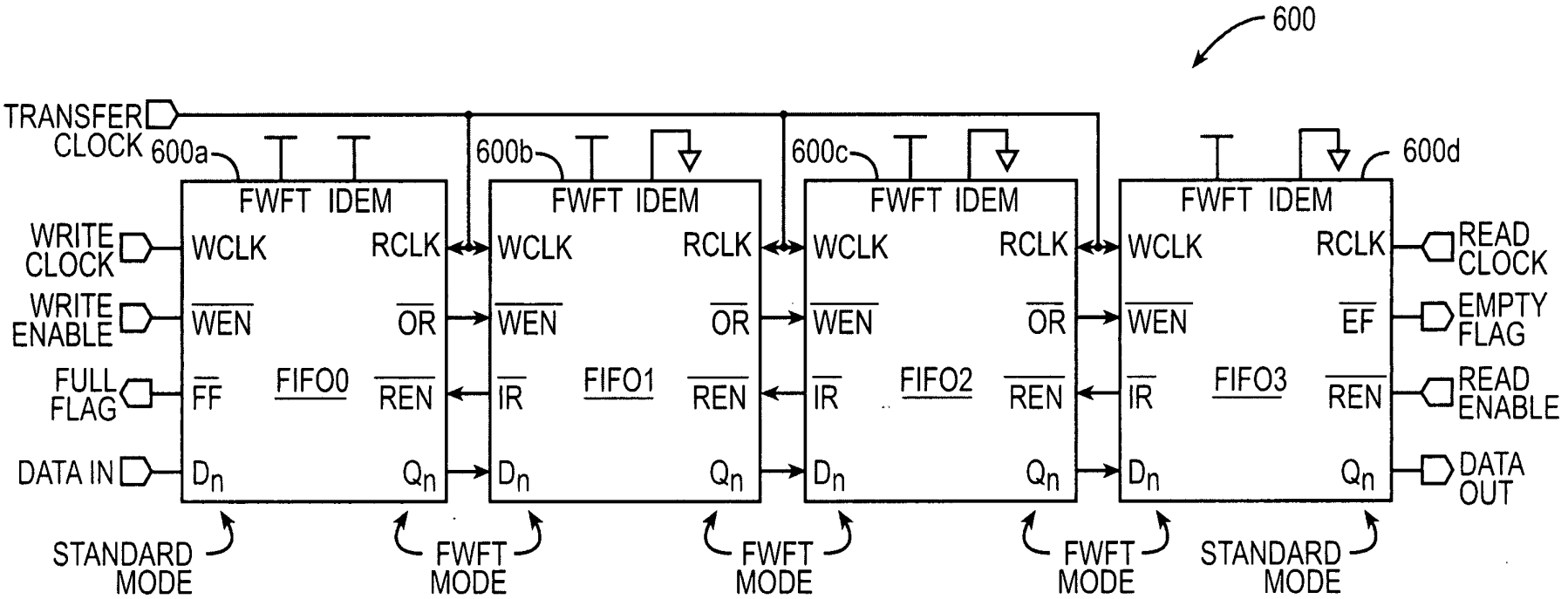

600
TRANSFER CLOCK
600a
600b
600c
600d
FWFT IDEM
WRITE CLOCK
WRITE ENABLE
FULL FLAG
DATA IN
WCLK
RCLK
WEN
OR
FF
FIFO0
REN
IR
FIFO1
FIFO2
FIFO3
Dn
Qn
EF
READ CLOCK
EMPTY FLAG
READ ENABLE
DATA OUT
STANDARD MODE
FWFT MODE

FIG. 9A

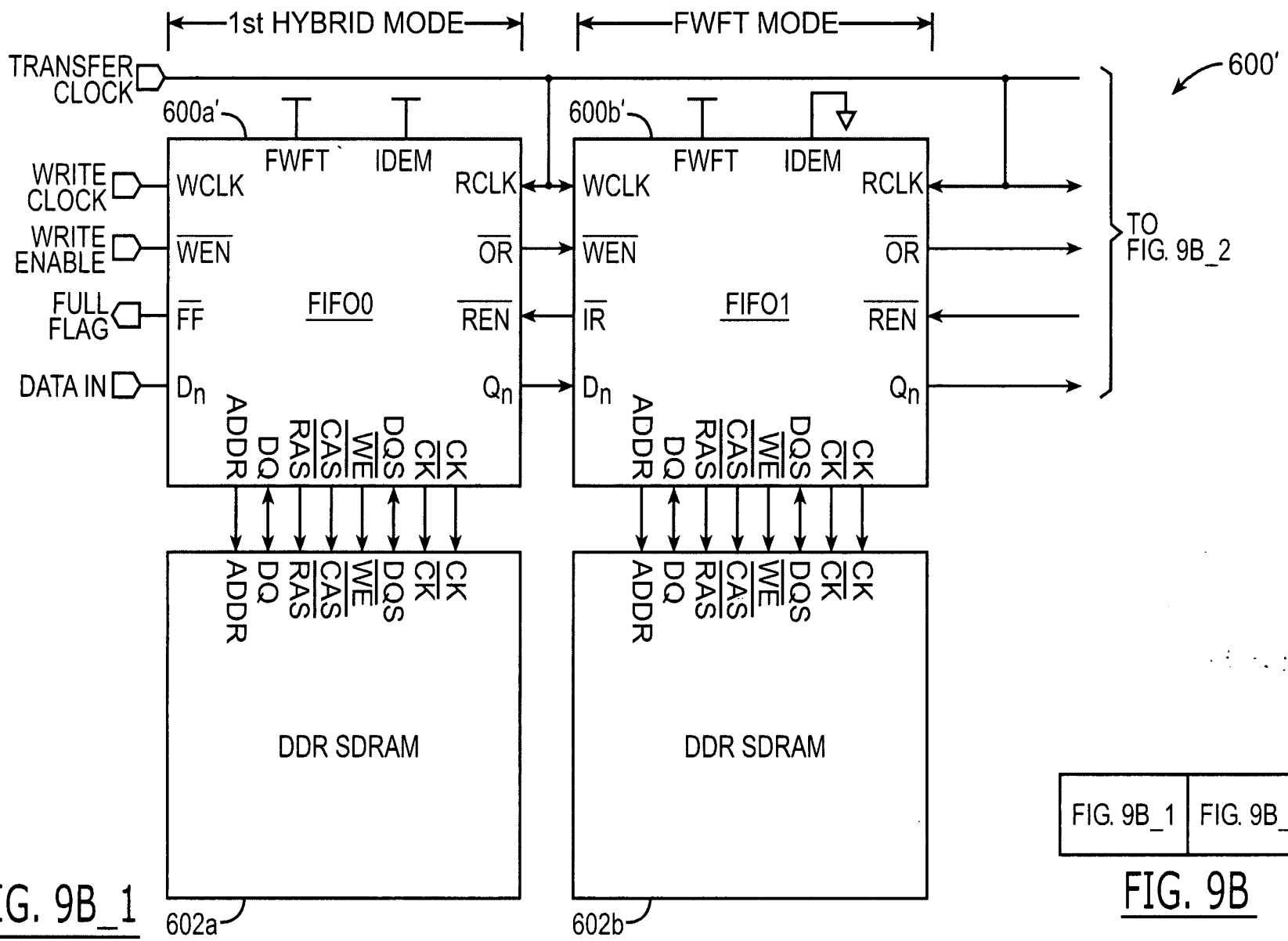



FIG. 9B_1

| FIG. 9B_1 | FIG. 9B_2 |
|---|---|

FIG. 9B

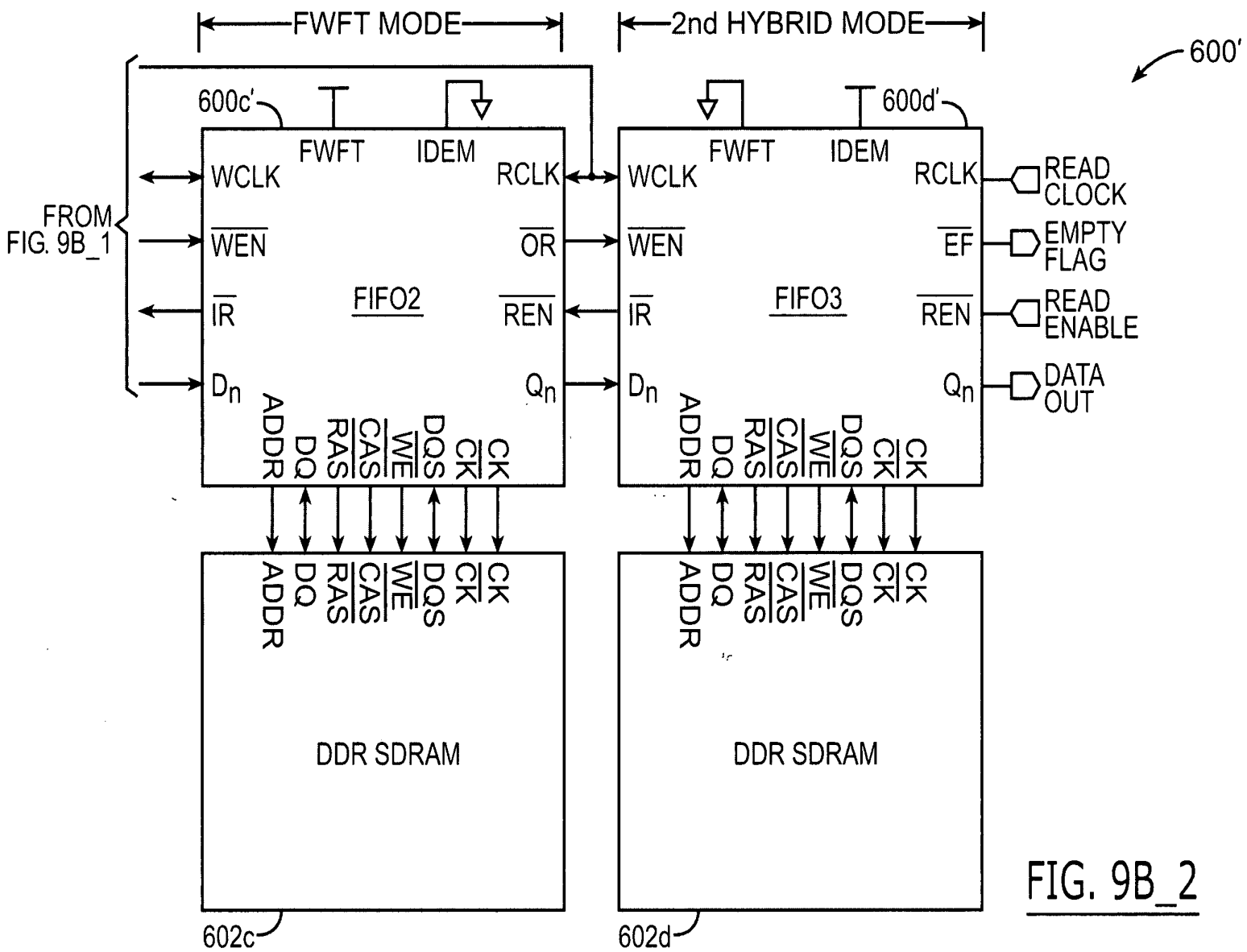

FWFT MODE
2nd HYBRID MODE
600'
600c'
600d'
FWFT
IDEM
FWFT
IDEM
FROM FIG. 9B_1
WCLK
WEN
IR
Dn
RCLK
OR
REN
Qn
FIFO2
WCLK
WEN
IR
Dn
RCLK
EF
REN
Qn
FIFO3
READ CLOCK
EMPTY FLAG
READ ENABLE
DATA OUT
ADDR
DQ
RAS
CAS
WE
DQS
CK
CK
DDR SDRAM
DDR SDRAM
602c
602d

FIG. 9B_2